

## **DETERMINISTIC NETWORK INTERDICTION**

R. Kevin Wood

Operations Research Department, Naval Postgraduate School  
Monterey, CA 93940, U.S.A.

**Abstract**—Interest in network interdiction has been rekindled because of attempts to reduce the flow of drugs and precursor chemicals through river and road networks in South America. This paper considers a problem in which an enemy attempts to maximize flow through a capacitated network while an interdictor tries to minimize this maximum flow by interdicting (stopping flow on) network arcs using limited resources. This problem is shown to be NP-complete even when the interdiction of an arc requires exactly one unit of resource. New, flexible integer programming models are developed for the problem and its variations, and valid inequalities and a reformulation are derived to tighten the LP relaxation of some of these models. A small computational example from the literature illustrates a hybrid (partly directed and partly undirected) model and the usefulness of the valid inequalities and the reformulation.

**Keywords:** Network interdiction, drug interdiction, maximum flow, integer programming, NP-complete, valid inequality

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>1993</b>		2. REPORT TYPE <b>N/A</b>		3. DATES COVERED <b>-</b>	
4. TITLE AND SUBTITLE <b>Deterministic Network Interdiction</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Naval Postgraduate School Operations Research Department Monterey, CA 93943</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release, distribution unlimited</b>					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <b>Interest in network interdiction has been rekindled because of attempts to reduce the flow of drugs and precursor chemicals through river and road networks in South America. This paper considers a problem in which an enemy attempts to maximize flow through a capacitated network while an interdictor tries to minimize this maximum flow by interdicting (stopping flow on) network arcs using limited resources. This problem is shown to be NP-complete even when the interdiction of an arc requires exactly one unit of resource. New, flexible integer programming models are developed for the problem and its variations, and valid inequalities and a reformulation are derived to tighten the LP relaxation of some of these models. A small computational example from the literature illustrates a hybrid (partly directed and partly undirected) model and the usefulness of the valid inequalities and the reformulation.</b>					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>SAR</b>	18. NUMBER OF PAGES <b>19</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

## DETERMINISTIC NETWORK INTERDICTION

R. KEVIN WOOD

Operations Research Department, Naval Postgraduate School  
Monterey, CA 93940, U.S.A.

**Abstract**—Interest in network interdiction has been rekindled because of attempts to reduce the flow of drugs and precursor chemicals moving through river and road networks in South America. This paper considers a problem in which an enemy attempts to maximize flow through a capacitated network while an interdictor tries to minimize this maximum flow by interdicting (stopping flow on) network arcs using limited resources. This problem is shown to be NP-complete even when the interdiction of an arc requires exactly one unit of resource. New, flexible, integer programming models are developed for the problem and its variations and valid inequalities and a reformulation are derived to tighten the LP relaxations of some of these models. A small computational example from the literature illustrates a hybrid (partly directed and partly undirected) model and the usefulness of the valid inequalities and the reformulation.

### 1. INTRODUCTION

Interest in the topic of network interdiction has been revived recently resulting from the U.S. anti-drug effort. In particular, the U.S. Army's SOUTHCOM, which directs U.S. anti-drug efforts in South America, would like to best allocate its limited resources to interdicting coca, partially processed cocaine and precursor chemicals in the South American drug producing areas [1]. Current emphasis is on interdicting the flow of precursor chemicals. Most of the traffic in these chemicals is carried along rivers in remote forested regions and along some inter-connecting roads. Consequently, in this situation, the interdiction problem can be viewed as a network interdiction problem. It is the purpose of this paper to review earlier deterministic network interdiction models, devise new solution techniques for these models, and develop new models and solution techniques. We believe that these models may be useful when combined with simulations for devising randomized strategies for interdicting drug and precursor chemical traffic. A subsequent paper will investigate probabilistic and game-theoretic models.

The simplest network interdiction problem arises from the well-known max flow-min cut theorem [2]. In this case, an enemy attempts to traverse from node  $s$  to node  $t$  in a directed network while the interdictor tries to break arcs in the network to eliminate all possible paths for the enemy. Each arc  $(i, j)$  has associated with it a resource expenditure  $r_{ij}$  which is required to break the arc and the interdictor wishes to use minimum total effort to disrupt all  $s$ - $t$  paths enemy paths. The solution to the problem is to let  $r_{ij}$  correspond to the capacity of arc  $(i, j)$ , find the maximum flow from  $s$  to  $t$  subject to the arc capacities and, using that solution, identify the minimum capacity cut. The arcs in the minimum capacity cut are those which should be broken to eliminate all enemy paths using minimum total effort.

One of the simplest variations on the above problem is to allow only a limited amount of resource to break arcs so as to leave as few as possible arc-disjoint paths remaining for the enemy to use. This paper will be primarily concerned with a generalization of this problem:

**PROBLEM 1.** An enemy wishes to move as much of a single commodity from node  $s$  to node  $t$  in a directed network. Each arc  $(i, j)$  has a capacity of  $u_{ij}$  units of commodity and requires of the interdictor an expenditure of  $r_{ij}$  units of resource to break the arc. Partially breaking an arc is not allowed. The problem for the interdictor is to minimize the maximum amount of flow the enemy can push through the network along unbroken arcs where the enemy is constrained by

Typeset by  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$

the arc capacities and the interdicator is constrained to use no more than the  $R$  units of resource available. It is assumed that all data are positive integers.

Problem 1 is very limited as stated but one of the strengths of our mathematical programming approach will be that generalizations are easy. Some generalizations that will be considered include multiple resources necessary to break an arc, multiple alternatives for breaking an arc using distinct resources, multiple sources and sinks, multiple commodities which are needed in specified proportions, and the ability of the interdicator to partially break an arc using a fraction of the resource needed to completely break the arc.

The basic problem, with minor variations, has been studied before in [3–5]. (Related work includes [6–10]). These works cite military applications, not drug interdiction. The approach used in [3] requires the network to be source-sink planar so that the network’s dual can be used, further requires the enumeration of many cuts and is not generalizable. It might be argued that a planarity assumption is not too restrictive in practice, although, it would tend to rule out adding air transportation arcs into the model. However, a source-sink planar network requires not only planarity but also requires that the source and sink lie on the outer face (exterior) of the network. This is a strong assumption, especially when a drug lab, i.e., a sink node for precursor chemicals, is likely to be found hidden in the middle of a network of rivers and roads. Furthermore, the enumeration of cuts is an exponential process in the worst case and a methodology based on this enumeration is unsatisfactory. The methodology used in [5] is limited in that it too requires the network to be source-sink planar but a dynamic programming approach is employed which avoids enumerating cuts and shows that the source-sink planar problem can be solved in pseudo-polynomial time. Generalizations as simple as having more than one distinct resource necessary to interdict an arc would be impossible except in very simple networks, however. (The difficulty here is analogous to the difficulty which arises when trying to generalize the dynamic programming solution of a knapsack problem to multiple resource constraints, i.e., exponential growth in complexity.) The methodology described in [4] does not require planarity and a specialized branch-and-bound algorithm is developed for the problem. However, that approach uses a very crude bounding procedure which would be difficult or impossible to generalize to multiple resources, multiple commodities, multiple sources and sinks, etc. The shortcomings of the aforementioned approaches invite a fresh look at the basic network interdiction problem.

In the following sections we first define a few terms and notation, consider the inherent complexity of Problem 1 and then propose an integer programming model whose solution we claim yields an answer to the problem. We then prove the validity of the model. Next, we describe a few extensions and variations on the basic model and then give a computational example previously found in the literature. Finally we consider a few open questions and make concluding remarks.

## 2. DEFINITIONS AND NOTATION

$G = (N, A)$  will denote a directed network with node set  $N$  and arc set  $A$ . We will usually refer to an arc as an ordered pair  $(i, j)$  where  $i, j \in N$ , although we can also refer to it by its number  $k$ . It is assumed that  $G$  contains no self loops, i.e., no arcs of the form  $(i, i)$ . If  $A' \subseteq A$ , then  $G - A'$  indicates  $G$  with edges  $A'$  deleted and if  $N' \subseteq N$ , then  $G - N'$  denotes  $G$  with all nodes in  $N'$  deleted along with all arcs incident into or from nodes in  $N'$ . It will be useful to distinguish two nodes  $s$  and  $t$  with  $s \neq t$ . Maximizing flow from  $s$  to  $t$  will be the same as maximizing the flow along an extra “return arc”  $(t, s)$  added to  $A$ .

An  $s$ - $t$  cutset is a partition of  $N$  into two subsets  $N_s$  and  $N_t$  such that  $s \in N_s$  and  $t \in N_t$ . With respect to that cut, an arc is a “forward” arc if it is directed from a node in  $N_s$  to a node in  $N_t$  and it is “backward” if it is directed from a node in  $T$  to a node in  $S$ . If each arc  $(i, j)$  has a capacity  $u_{ij}$  then the capacity of the cut is the sum of the capacities of forward arcs associated with the cut.

$G = (N, A)$  may also denote an undirected network. For an undirected network the definition of a cut is analogous to that in a directed network and the capacity of the cut is the sum of the capacities of the arcs with one endpoint in  $N_s$  and the other endpoint in  $N_t$ . In some cases it will be necessary to refer to an undirected arc  $(i, j) \in A$  and its representation as a directed arc  $(i, j)$  and a directed arc  $(j, i)$  in anti-parallel. Let  $A'$  denote the set of directed arcs  $(i, j)$  such

that  $i < j$  and  $(i, j) \in A$ , and let  $A''$  denote the set of directed arcs  $(j, i)$  such that  $(i, j) \in A'$ . Thus, each arc in  $A$  is represented once in  $A'$  oriented in one direction and is represented once in  $A''$  oriented in the opposite direction.

In proving some complexity results we will also want to consider an undirected graph  $H = (V, E)$  where  $V$  is the set of vertices and  $E$  the set of edges, i.e., unordered pairs of vertices. The degree of a vertex  $v$ , denoted  $\deg(v)$ , is the number of edges of the form  $(u, v)$ . A self loop is an edge of the form  $(v, v)$ .  $H$  is a tree if it is connected and has no cycles. If  $E' \subseteq E$ , then  $H - E'$  denotes  $H$  with all edges in  $E'$  deleted.

### 3. COMPLEXITY

In this section we show that the decision problems associated with Problem 1 and two variants are NP-complete. We will first show a simple transformation of the binary knapsack problem (decision) to Problem 1 (decision), which are defined as follows:

**BINARY KNAPSACK PROBLEM (decision).** Given: A set of items  $K$  with each item  $k \in K$  having a positive integer profit  $u'_k$  and a positive integer weight  $r'_k$ , and two positive integers  $U'$  and  $R'$ . Question: Does there exist a subset  $K' \subset K$  such that  $\sum_{k \in K'} u'_k \geq U'$  and  $\sum_{k \in K'} r'_k \leq R'$ , i.e., does there exist a set of items whose total profit is at least  $U'$  and whose total weight is no more than  $R'$ ?

**PROBLEM 1 (decision).** Given: A directed graph  $G = (N, A)$  with distinguished nodes  $s$  and  $t$ , positive integer capacities  $u_k$  for each arc  $k \in A$  and positive integer resource  $r_k$  required for deletion of any arc  $k \in A$  and two positive integers  $U$  and  $R$ .

Question: Does there exist a subset of arcs  $A' \subset A$  such that  $\sum_{k \in A'} r_k \leq R$  and the maximum  $s$ - $t$  flow in  $G - A'$  is no more than  $U$ , i.e., does there exist a subset of arcs whose deletion consumes no more than  $R$  units of resource and which leaves behind a network with maximum  $s$ - $t$  flow not exceeding  $U$ ?

**THEOREM 1.** *Problem 1 (decision) is NP-complete.*

**PROOF.** Consider a knapsack problem as defined above, which is well-known to be NP-complete (e.g., [11]). Now create a directed network  $G = (N, A)$  with two nodes  $s$  and  $t$  and for each item  $k \in K$  in the knapsack problem create an arc  $k \in A$  directed from  $s$  to  $t$  with capacity  $u_k = u'_k$  and resource requirement  $r_k = r'_k$ . Furthermore, define  $R = R'$  and  $U = \sum_k u_k - U'$ . Now, suppose there exists a subset  $K' \subset K$  such that  $\sum_{k \in K'} u'_k \geq U'$  and  $\sum_{k \in K'} r'_k \leq R'$ . Let  $A'$  correspond to  $K'$ . Then it follows that, because of the simple topology of the network, that the maximum flow in  $G - A'$  is at most  $\sum_k u_k - U' = U$  and trivially  $\sum_{k \in A'} r_k \leq R$ . Conversely, suppose there exists a set of arcs  $A'$  in  $G$  such that the maximum  $s$ - $t$  flow in  $G - A'$  is no more than  $U$  and  $\sum_{k \in A'} r_k \leq R$ . Then, letting  $K'$  in the knapsack problem correspond to  $A'$ , it follows that  $\sum_{k \in K'} u'_k \geq \sum_k u_k - U = U'$  and, trivially,  $\sum_{k \in A'} r'_k \leq R$ . Together with the fact that Problem 1 (decision) is clearly in NP, this implies that Problem 1 is NP-complete. ■

Note that Problem 1 specialized to planar networks or undirected networks is still NP-complete since the network created in the proof is planar and could equally well have been undirected. The proof does not, however, show that the problem is NP-complete in the strong sense. That is, the proof leaves open the possibility of a pseudo-polynomial time algorithm for solving Problem 1. To show that, in fact, Problem 1 is strongly NP-complete, we show that the following specialization of Problem 1 is strongly NP-complete.

**PROBLEM 2.** This is the same as Problem 1 except that interdiction of an arc requires exactly one unit of resource.

So in Problem 2 the interdictor just has a cardinality constraint on the number of arcs he can break rather than a general resource constraint. The proof used for Theorem 1 does not follow through for Problem 2 since a knapsack problem in which each item weighs exactly one unit is trivial to solve. To prove Problem 2 is strongly NP-complete, we state the problem as a decision problem:

**PROBLEM 2 (decision).** Given: Directed graph  $G = (N, A)$  with distinguished nodes  $s$  and  $t$ , positive integer capacities of  $u_{ij}$  for each arc  $(i, j) \in A$ , a positive integer  $R$ , and positive integer  $U$ .

Question: Does there exist a set of arcs  $A'$  with  $|A'| \leq R$  such that the maximum  $s$ - $t$  flow in  $G - A'$  has value  $U$ ?

We will show a transformation from the NP-complete problem “clique” [11]:

**PROBLEM CLIQUE** (decision). Given: An undirected graph  $H = (V, E)$  and positive integer  $K$ .

Question: Does there exist a subgraph of  $H$  which is a clique (complete graph) on  $K$  vertices?

In proving that Problem 2 is NP-complete, we will define a capacitated directed network  $\hat{G}^H$  with respect to graph  $H$  and show that there exists a set of  $R$  arcs  $\hat{A}'$  in  $\hat{G}^H$  such that  $\hat{G}^H - \hat{A}'$  has a maximum flow of  $U = K$  units if and only if  $H$  contains a clique on  $K$  vertices.  $R$  will be  $|E| - \binom{K}{2}$ . To do this, we prove several lemmas which use an intermediate construct illustrated in Figure 1. We are given an undirected graph  $H = (V, E)$ , which, without loss of generality, has no parallel edges and no self loops. From  $H$  we create a capacitated, directed graph  $G^H = (N, A)$  as follows: For each edge  $e \in E$  create a node  $i_e$  in a node set  $N_1$  and for each vertex  $v \in V$  create a node  $j_v$  in a node set  $N_2$ . In addition, create special nodes  $s$  and  $t$ . Now, for each edge  $e \in E$ , direct an arc in  $G^H$  from  $s$  to  $i_e$  with capacity 2 and call this set of arcs  $A_1$ . Next, for each edge  $e = (u, v)$  direct an arc in  $G$  with capacity 1 from  $i_e$  to  $j_u$  and direct another arc with capacity 1 from  $i_e$  to  $j_v$ . Let this be the set of arcs  $A_2$ . Finally, for each vertex  $v \in V$  direct an arc with capacity 1 from  $j_v$  to  $t$  and call this arc set  $A_3$ . This completes the construction of  $G^H = (N, A) = (\{s\} \cup \{t\} \cup N_1 \cup N_2, A_1 \cup A_2 \cup A_3)$ .

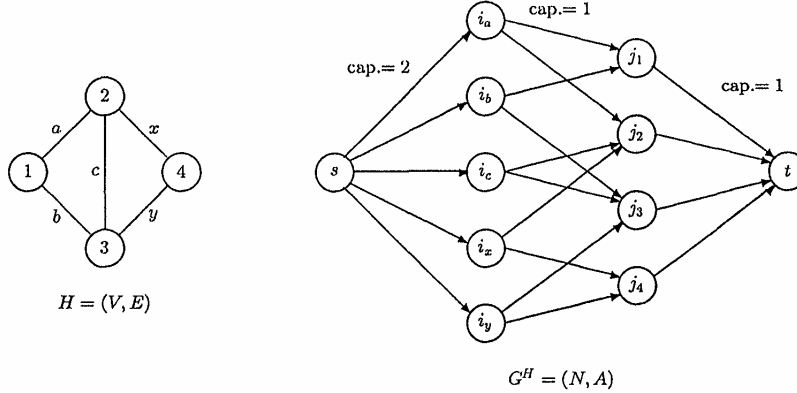


Figure 1. Illustration for proofs of Lemmas 1 and 2.

**LEMMA 1.** *The maximum flow in  $G^H$  constructed from  $H$  as above is equal to the number of vertices  $v \in V$  with  $\deg(v) > 0$ .*

**PROOF.** In other words, let  $V_0$  be the subset of  $V$  containing all vertices having degree 0; then, we wish to show that the maximum flow in  $G^H$  is  $|V - V_0|$ . Suppose that  $H$  is a tree. Select an edge  $e = (u, v)$  such that  $\deg(v) = 1$ . (Two such edges must always exist in a tree.) One unit of flow in  $G^H$  can be routed from  $s$  to  $i_e$  to  $j_v$  to  $t$ . Now delete  $e$  and  $v$  from  $H$  to create another tree, find another edge  $e = (u, v)$  with  $\deg(v) = 1$ , route another unit of flow from  $s$  to the new  $i_e$  to the new  $j_v$  to  $t$  and repeat until  $H$  consists of a single edge and its incident vertices. At each step of the process, it is possible to route another unit of flow through  $G^H$  because the path using arcs  $(s, i_e)$ ,  $(i_e, j_v)$  and  $(j_v, t)$  has not been used before. Then, with one edge  $e = (u, v)$  remaining, we can route one unit of flow from  $s$  to  $i_e$  to  $j_v$  to  $t$ , and because the capacity of  $(s, i_e)$  is 2, we can also route one unit of flow from  $s$  to  $i_e$  to  $j_u$  to  $t$ . Since a tree has  $|V| - 1$  edges we have found a flow of  $|V|$  units from  $s$  to  $t$  and this is obviously maximum.

Now, consider a general graph  $H = (V, E)$ . Clearly, the maximum flow in  $G^H$  is at most  $|V - V_0|$  since, if  $\deg(v) = 0$ , it is not possible to route any flow through  $j_v$  to  $t$ . Now let  $H_k = (V_k, E_k)$  denote a connected component of  $H$  such that  $|E_k| > 0$ , i.e.,  $\deg(v) > 0$  for all  $v \in V_k$ . Since  $H_k$  contains a tree we can route one unit of flow (and obviously only one) through each node  $j_v$  to  $t$  for each vertex  $v \in V_k$ . Thus the maximum flow in  $H$  equals  $|\cup_k V_k| = |V - V_0|$ . ■

LEMMA 2. Let  $G^H$  be constructed from  $H$  as above. Then, there exists a set of arcs  $A'_1 \subseteq A_1$  with  $|A'_1| = |E| - \binom{K}{2}$  such that the maximum flow from  $s$  to  $t$  in  $G^H - A'_1$  is  $K$  if and only if  $G$  contains a clique of size  $K$ .

PROOF. ( $\Leftarrow$ ) If  $G$  contains a clique of size  $K$ , let  $E'$  be the set of edges not in the clique and let  $A'_1 \equiv \{(s, i_e) \mid e \in E'\}$ . Of necessity,  $|A'_1| = |E'| = |E| - \binom{K}{2}$ . Then, by construction, the maximum flow in  $G^H - A'_1$  is the same as the maximum flow in  $G^{H-E'}$ . (If  $(s, i_e)$  created from  $e = (u, v)$  is deleted from  $G^H$ , we might as well delete  $i_e$  and edges  $(i_e, j_u)$  and  $(i_e, j_v)$ , since no flow from  $s$  to  $t$  can be routed through them. If we make all such deletions,  $G^H - A'_1$  is transformed into  $G^{H-E'}$ .) But by Lemma 1, the maximum flow in  $G^{H-E'}$  is  $K$ , since  $H - E'$  has  $K$  vertices with degree  $K - 1 > 0$  and  $|V| - K$  vertices with degree 0.

( $\Rightarrow$ ) Next, suppose that it is possible to find a set of edges  $A'_1 \subseteq A_1$  with  $|A'_1| = |E| - \binom{K}{2}$  such that the maximum  $s$ - $t$  flow in  $G - A'_1$  is  $K$ . Let  $E'$  correspond to  $A'_1$  and let  $N'_1 \equiv \{i_e \mid i_e \in N_1, e \in E'\}$ . Then, by construction  $G^H - A'_1$  has the same maximum flow as  $G^H - N'_1$  and  $G^H - N'_1 = G^{H-E'}$ . Since  $G^H - N'_1$  has a maximum flow of  $K$ , it follows that  $H - E'$  has  $K$  vertices  $v$  with  $\deg(v) > 0$  and  $|V| - K$  vertices  $v$  with  $\deg(v) = 0$ . Therefore,  $H - E'$  has a total of  $\binom{K}{2}$  edges (none of which are in parallel) which are incident to exactly  $K$  vertices. The only way this can occur is if  $H - E'$  consists of a complete graph on  $K$  vertices and  $|V| - K$  isolated vertices. Thus,  $H$  contains a clique of size  $K$ . ■

THEOREM 2. Problem 2 is NP-complete.

PROOF. Problem 2 (decision) is clearly in NP. For any instance of Problem Clique with graph  $H = (V, E)$ , we will create a capacitated directed network  $\hat{G}^H$  such that  $H$  contains a clique on  $K$  vertices if and only if  $\hat{G}^H$  contains a set of arcs  $\hat{A}'$  with  $|\hat{A}'| = R$  such that  $\hat{G}^H - \hat{A}'$  has a maximum  $s$ - $t$  flow of  $K$ . To avoid the tedious details of trivial cases, we assume that  $|E| \geq 1$  and  $K \geq 2$ . Create  $\hat{G}^H$  by first creating  $G^H$  from  $H$  as described before. Then, replace each arc  $(i_e, j_v)$  with  $|E|$  parallel arcs each with capacity  $1/|E|$  and call this arc set  $\hat{A}_2$ . Do the same thing for arcs of the form  $(j_v, t)$  and call that new arc set  $\hat{A}_3$ . Then  $\hat{G}^H = (N, \hat{A}) = (\{s\} \cup \{t\} \cup N_1 \cup N_2, A_1 \cup \hat{A}_2 \cup \hat{A}_3)$ . Also, let  $R = |E| - \binom{K}{2}$ .

( $\Leftarrow$ ) Let  $E'$  be the edges of  $E$  not in the clique and let  $\hat{A}' = \{(s, i_e) \mid e \in E'\}$ . Then,  $|\hat{A}'| = R$  and, because of the similarity in structure between  $G^H$  and  $\hat{G}^H$ , it follows from Lemma 2 that the maximum flow in  $\hat{G}^H - \hat{A}'$  is  $K$ .

( $\Rightarrow$ ) We are given a set of  $R$  arcs  $\hat{A}'$  such that the maximum flow in  $\hat{G}^H - \hat{A}'$  is  $K$ . If  $\hat{A}' \subseteq A_1$ , it follows immediately from Lemma 2 that  $H$  contains a clique of size  $K$ , because of the similarity in structure between  $G^H$  and  $\hat{G}^H$ . Now suppose that  $\hat{A}' = \hat{A}'_1 \cup \hat{A}'_2 \cup \hat{A}'_3$  where  $\hat{A}'_2 \cup \hat{A}'_3 \neq \emptyset$ . We will show that the maximum flow in  $\hat{G}^H - \hat{A}'$  must be strictly greater than  $K$ , which implies that  $\hat{A}'$  cannot contain any arcs from  $\hat{A}_2$  or  $\hat{A}_3$  and the proof will be complete. Since  $|\hat{A}'_2 \cup \hat{A}'_3| \geq 1$ , it must be that  $A_1 - \hat{A}'_1$  contains at least  $\binom{K}{2} + 1$  arcs and using the same reasoning as in Lemma 2,  $\hat{G}^H - \hat{A}'_1$  has a maximum flow of at least  $K + 1$ . But  $\hat{G}^H - \hat{A}'$  can be created from  $\hat{G}^H - \hat{A}'_1$  by deleting no more than  $|E| - \binom{K}{2} \leq |E| - 1$  additional arcs taken from  $\hat{A}_2 \cup \hat{A}_3$ . Each such deletion from  $\hat{G}^H - \hat{A}'_1$  reduces the maximum flow by at most  $1/|E|$ , since the capacities of these arcs are  $1/|E|$ . Thus, the maximum flow in  $\hat{G}^H - \hat{A}'$  is no less than  $K + 1 - (|E| - 1)(1/|E|) = K + 1/|E|$ . ■

COROLLARY 1. Problems 1 and 2 are NP-complete in the strong sense.

PROOF. Problem 2 is NP-complete in the strong sense since the proof of its NP-completeness did not require any entities whose number was dependent on the numerical values  $u_{ij}$ . Problem 1 is therefore NP-complete in the strong sense because it can be specialized to Problem 2. ■

The proof of Theorem 2 uses a possibly non-planar graph so that proof does not imply that Problem 2 is NP-complete for planar networks. Indeed, this problem can be solved in polynomial time for “ $s$ - $t$  planar graphs” [6]. Likewise, the proof does not imply that Problem 1 is NP-complete in the strong sense for planar networks which is fortunate since a pseudo-polynomial time algorithm for this problem is described in [5]. Finally, we note that if Problem 2 is further specialized so that  $u_{ij} = 1$  for all arcs  $(i, j)$ , the resulting problem is of polynomial complexity:

Find a minimum cardinality cut in polynomial time using a maximum flow algorithm and then break  $R$  arcs in that cut.

#### 4. AN INTEGER PROGRAMMING SOLUTION

In this section, we give an integer programming model which will solve Problem 1. We first state the model and then prove its correctness starting with a formal, "min-max" formulation of Problem 1.

The formulation of the model is:

MODEL 1D

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} u_{ij} \beta_{ij}, \\ \text{s.t.} \quad & \alpha_i - \alpha_j + \beta_{ij} + \gamma_{ij} \geq 0, \quad \forall (i,j) \in A, \\ & \alpha_t - \alpha_s \geq 1, \end{aligned} \tag{1}$$

$$\sum_{(i,j) \in A} r_{ij} \gamma_{ij} \leq R, \tag{2}$$

$$\begin{aligned} \alpha_i &\in \{0, 1\}, \quad \forall i \in N, \\ \beta_{ij}, \gamma_{ij} &\in \{0, 1\}, \quad \forall (i,j) \in A. \end{aligned}$$

"D" is used to denote that this model is for a directed network. Model 1D is based on a modified dual of the max flow linear programming formulation. Essentially, an  $s$ - $t$  cut is identified with all  $\alpha_i = 1$  for  $i$  on the  $t$  side of the cut and  $\alpha_i = 0$  for all  $i$  on the  $s$  side of the cut. The value of  $\gamma_{ij}$  is 1 if  $(i,j)$  is a forward arc across the cut which is to be broken;  $\beta_{ij}$  is 1 if  $(i,j)$  is a forward arc across the cut but it is not to be broken; and all other  $\beta_{ij}$  and  $\gamma_{ij}$  are 0. Thus, we see that a cut is identified and arcs are broken in that cut so as to leave as little remaining capacity as possible.

In order to prove that the solution of Model 1D solves Problem 1 it is possible to proceed from the fact that the max flow problem is totally unimodular, and, consequently its dual is totally unimodular (e.g., [12]). The approach taken here seems more direct, however. We need the following lemma.

LEMMA 3. *The dual of the max flow problem has an optimal solution in which all variables are 0 or 1.*

PROOF. The max flow problem is

$$\begin{aligned} \max \quad & x_{ts}, \\ \text{s.t.} \quad & \sum_j x_{sj} - \sum_j x_{js} - x_{ts} = 0, \\ & \sum_j x_{ij} - \sum_j x_{ji} = 0, \quad \forall i \in N - \{s, t\}, \\ & \sum_j x_{tj} - \sum_j x_{jt} + x_{ts} = 0, \\ & 0 \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A, \\ & x_{ts} \geq 0, \end{aligned}$$

where  $x_{ts}$  corresponds to the return arc which has been added to the network going from  $t$  to  $s$ . The dual of the max flow problem is

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} u_{ij} \theta_{ij}, \\ \text{s.t.} \quad & \alpha_i - \alpha_j + \theta_{ij} \geq 0, \quad \forall (i,j) \in A, \end{aligned} \tag{3}$$

$$\begin{aligned} & \alpha_t - \alpha_s \geq 1, \tag{4} \\ & \theta_{ij} \geq 0, \quad \forall (i,j) \in A. \end{aligned}$$



Let  $(N_s, N_t)$  correspond to the minimum capacity cut in  $G$ . Let  $\alpha_i = 1$  for all  $i \in N_t$ , let  $\alpha_i = 0$  for all  $i \in N_s$ , let  $\theta_{ij} = 1$  for all arcs  $(i, j)$  which are forward arcs in the cut and let all other  $\theta_{ij} = 0$ . Clearly, equation (4) is satisfied and equations (3) can be seen to be satisfied by checking against the four classes of arcs  $(i, j)$ :

- (a)  $i \in N_s, j \in N_s$ ,
- (b)  $i \in N_s, j \in N_t$ ,
- (c)  $i \in N_t, j \in N_s$ , and
- (d)  $i \in N_t, j \in N_t$ .

Thus, this solution is feasible. Furthermore, it is optimal because the value of the objective function equals the capacity of this cut which is an upper bound on the maximum flow, but the maximum flow equals the capacity of the cut by linear programming duality. ■

We can now prove the correctness of Model 1D.

**THEOREM 3.** *The solution to Model 1D solves Problem 1.*

**PROOF.** A model to solve Problem 1 can be formally stated as the following min-max flow-based model.

**MODEL 2D**

$$\begin{aligned}
 & \min_{\gamma \in \Gamma} \max_x x_{ts}, \\
 & \text{s.t.} \quad \sum_j x_{sj} - \sum_j x_{js} - x_{ts} = 0, \\
 & \quad \sum_j x_{ij} - \sum_j x_{ji} = 0, \quad \forall i \in N - \{s, t\}, \\
 & \quad \sum_j x_{tj} - \sum_j x_{jt} + x_{ts} = 0, \\
 & \quad x_{ij} - u_{ij}(1 - \gamma_{ij}) \leq 0, \quad \forall (i, j) \in A, \\
 & \quad x_{ij} \geq 0, \quad \forall (i, j) \in A \cup \{(t, s)\},
 \end{aligned} \tag{5}$$

where  $\Gamma \equiv \{\gamma_{ij} \mid \gamma_{ij} \in \{0, 1\} \forall (i, j) \in A, \sum_{(i, j) \in A} u_{ij} \gamma_{ij} \leq R\}$ .

Now, for fixed  $\gamma_{ij}$  the dual of the inner maximization problem can be taken giving the equivalent model:

**MODEL 3D**

$$\begin{aligned}
 & \min_{\gamma \in \Gamma} \min_{\alpha, \theta} \sum_{(i, j) \in A} u_{ij}(1 - \gamma_{ij})\theta_{ij}, \\
 & \text{s.t.} \quad \alpha_i - \alpha_j + \theta_{ij} \geq 0, \quad \forall (i, j) \in A, \\
 & \quad \alpha_t - \alpha_s \geq 1, \\
 & \quad \alpha_i \in \{0, 1\}, \quad \forall i \in N, \\
 & \quad \theta_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A.
 \end{aligned}$$

The  $\alpha_i$  and  $\theta_{ij}$  values can be restricted to 0 or 1 because, for fixed values of  $\gamma_{ij}$ , the inner maximization of Model 2D is just a max flow problem and Lemma 1 applies. Next, Model 3D can be linearized by replacing  $(1 - \gamma_{ij})\theta_{ij}$  with  $\beta_{ij}$  where  $\beta_{ij} \in \{0, 1\}$  and  $\beta_{ij} \geq \theta_{ij} - \gamma_{ij}$ . This yields:

**MODEL 4D**

$$\begin{aligned}
 & \min \sum_{(i, j) \in A} u_{ij} \beta_{ij}, \\
 & \text{s.t.} \quad \alpha_i - \alpha_j + \theta_{ij} \geq 0, \quad \forall (i, j) \in A, \\
 & \quad \alpha_t - \alpha_s \geq 1,
 \end{aligned}$$

$$\beta_{ij} + \gamma_{ij} - \theta_{ij} \geq 0, \quad \forall (i, j) \in A, \quad (6)$$

$$\sum_{(i,j) \in A} r_{ij} \gamma_{ij} \leq R, \quad (7)$$

$$\begin{aligned} \alpha_i &\in \{0, 1\}, & \forall i \in N, \\ \theta_{ij} &\in \{0, 1\}, & \forall (i, j) \in A, \\ \beta_{ij}, \gamma_{ij} &\in \{0, 1\}, & \forall (i, j) \in A. \end{aligned} \quad (8)$$

Next we argue that constraints (6) can be replaced by equalities. Consider any optimal solution in which  $\theta_{ij} = 0$ . For feasibility  $\beta_{ij}$  and  $\gamma_{ij}$  can be either 0 or 1. But setting  $\beta_{ij}$  to 1 would unnecessarily increase the value of the objective function so it must be 0. It is possible that  $\gamma_{ij} = 1$  in some optimal solution but then setting  $\gamma_{ij}$  to 0 creates another optimal solution because it does not change the value of the objective function and maintains feasibility. Thus, if  $\theta_{ij} = 0$  it may be assumed that  $\beta_{ij} = \gamma_{ij} = 0$ , or equivalently,  $\beta_{ij} + \gamma_{ij} - \theta_{ij} = 0$ . Now suppose that in an optimal solution  $\theta_{ij} = 1$ . In that case, to have feasibility, either  $\beta_{ij} = 1$  or  $\gamma_{ij} = 1$  or both. Suppose that  $\beta_{ij} = 1$ . Then, the constraint  $\beta_{ij} + \gamma_{ij} - \theta_{ij} \geq 0$  is satisfied and, if  $\gamma_{ij} = 1$ , there exists another optimal solution in which  $\gamma_{ij} = 0$ . Thus, it may be assumed that, when  $\theta_{ij} = 1$ , either  $\beta_{ij} = 1$  or  $\gamma_{ij} = 1$  but not both, or equivalently  $\beta_{ij} + \gamma_{ij} - \theta_{ij} = 0$ . Thus, whether  $\theta_{ij} = 1$  or  $\theta_{ij} = 0$ , we have  $\beta_{ij} + \gamma_{ij} - \theta_{ij} = 0$ , which implies that inequalities of constraints (6) may be replaced by equalities.

Since  $\beta_{ij} + \gamma_{ij} - \theta_{ij} = 0$ , or  $\theta_{ij} = \beta_{ij} + \gamma_{ij}$ ,  $\beta_{ij} + \gamma_{ij}$  can be substituted everywhere for  $\theta_{ij}$ . This eliminates constraints (6) and (8) and yields Model 1D. ■

As a final point, note that constraint (1) can be eliminated by replacing  $\alpha_i$  by 1 and  $\alpha_s$  by 0 wherever they appear.

## 5. EXTENSIONS AND VARIANTS

In this section, we discuss number of modifications and extensions to the basic model which underscore the flexibility of the mathematical programming approach.

### 5.1. Cardinality Constraint

Problem 2, in which there is a cardinality constraint on the number of arcs interdicted rather than a general resource constraint, can be solved by solving Model 1D in which constraint (2) is replaced by  $\sum_{(i,j) \in A} \gamma_{ij} \leq R$ . We will refer to this model as Model 1DC.

### 5.2. Partial Arc Interdiction

In this scenario, we assume that by applying  $f_{ij}r_{ij}$  units of resource to arc  $(i, j)$  where  $0 \leq f_{ij} \leq 1$ , we can reduce the capacity of the arc to  $(1 - f_{ij})u_{ij}$ . Create Model 2P for this problem which is the same as Model 2D except that  $\Gamma \equiv \{\gamma_{ij} \mid 0 \leq \gamma_{ij} \leq 1 \forall (i, j) \in A, \sum_{(i,j) \in A} r_{ij} \gamma_{ij} \leq R\}$ . Then, making transformations analogous to those used to develop Model 1D, we obtain:

MODEL 1P

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} u_{ij} \beta_{ij}, \\ \text{s.t.} \quad & \alpha_i - \alpha_j + \beta_{ij} + \gamma_{ij} \geq 0, & \forall (i, j) \in A, \\ & \alpha_t - \alpha_s \geq 1, \\ & \sum_{(i,j) \in A} r_{ij} \gamma_{ij} \leq R, \\ & \alpha_i \in \{0, 1\}, & \forall i \in N, \\ & 0 \leq \gamma_{ij} \leq 1 & \forall (i, j) \in A, \\ & 0 \leq \beta_{ij} \leq 1, & \forall (i, j) \in A. \end{aligned}$$

### 5.3. Multiple Sources and Sinks

The standard way to handle multiple sources and sinks would be to create a super-source  $s$  and connect it to the individual sources  $s' \in N^S$  with infinite capacity, unbreakable arcs and to create a super-sink  $t$  and connect the individual sinks  $t' \in N^T$ , to it with infinite capacity, unbreakable arcs. However, the fact that these arcs are unbreakable and have infinite capacity implies that in the optimal solution the super-source will be on the same side of the optimal cut as the individual sources and the super-sink will be on the same side of the cut as the individual sinks. Thus, we know that  $\alpha_{s'} = 0$  for all  $s' \in N^S$  and  $\alpha_{t'} = 1$  for all  $t' \in N^T$ . Therefore, Model 1D can be modified to handle multiple sources and sinks by eliminating constraint (1) and substituting a 0 for every  $\alpha_{s'}$  such that  $s' \in N^S$  and substituting a 1 for every  $\alpha_{t'}$  such that  $t' \in N^T$ .

### 5.4. Undirected Networks

Next, we consider the analog of Problem 1 for undirected networks. We use the following proposition.

**PROPOSITION 1.** *The max-flow min-cut theorem holds for undirected networks as well as directed networks.*

The min capacity cut identification model for an undirected network can be most succinctly stated using  $A'$  and  $A''$  as defined with respect to an undirected network  $G = (N, A)$  in Section 2:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A'} u_{ij} \theta_{ij}, \\ \text{s.t.} \quad & \alpha_i - \alpha_j + \theta_{ij} \geq 0, \quad \forall (i,j) \in A', \\ & \alpha_j - \alpha_i + \theta_{ij} \geq 0, \quad \forall (i,j) \in A', \\ & \alpha_t - \alpha_s \geq 1. \end{aligned}$$

It is easy to verify that, associated with the min capacity cut  $(N_s, N_t)$ , there is a feasible solution to the above model in which  $\alpha_i = 0$  for all  $i \in N_s$ ,  $\alpha_i = 1$  for all  $i \in N_t$ ,  $\theta_{ij} = 1$  if  $i \in N_s$  and  $j \in N_t$  or  $j \in N_s$  and  $i \in N_t$ , and otherwise  $\theta_{ij} = 0$ . Furthermore, the value of the objective function equals the capacity of the cut. That this solution is optimal follows from Proposition 1, the fact that the dual of the above model is a max flow model for an undirected network:

$$\begin{aligned} \max_x \quad & x_{ts}, \\ \text{s.t.} \quad & \sum_{j:(s,j) \in A' \cup A''} x_{sj} - \sum_{j:(j,s) \in A' \cup A''} x_{js} - x_{ts} = 0, \\ & \sum_{j:(i,j) \in A' \cup A''} x_{ij} - \sum_{j:(j,i) \in A' \cup A''} x_{ji} = 0, \quad \forall i \in N - \{s, t\}, \\ & \sum_{j:(t,j) \in A' \cup A''} x_{tj} - \sum_{j:(j,t) \in A' \cup A''} x_{jt} + x_{ts} = 0, \\ & x_{ij} + x_{ji} \leq u_{ij}, \quad \forall (i,j) \in A', \\ & x_{ij} \geq 0, \quad \forall (i,j) \in A' \cup A'' \cup \{(t,s)\}, \end{aligned}$$

and duality. This all leads to the analog of Lemma 1:

**LEMMA 4.** *The dual of the max flow problem for undirected networks has a solution in which all variables are 0 or 1.*

The solution to Problem 1 for undirected networks can be found by solving the following min-max model:

MODEL 2U

$$\begin{aligned}
& \min_{\gamma \in \Gamma} \max_x x_{ts}, \\
& \text{s.t.} \quad \sum_{j:(s,j) \in A' \cup A''} x_{sj} - \sum_{j:(j,s) \in A' \cup A''} x_{js} = 0, \\
& \quad \sum_{j:(i,j) \in A' \cup A''} x_{ij} - \sum_{j:(j,i) \in A' \cup A''} x_{ji} = 0, \quad \forall i \in N - \{s, t\}, \\
& \quad \sum_{j:(t,j) \in A' \cup A''} x_{tj} - \sum_{j:(j,t) \in A' \cup A''} x_{jt} = 0, \\
& \quad x_{ij} + x_{ji} - u_{ij}(1 - \gamma_{ij}) \leq 0, \quad \forall (i, j) \in A', \\
& \quad x_{ij} \geq 0, \quad \forall (i, j) \in A' \cup A'' \cup \{(t, s)\},
\end{aligned}$$

where  $\Gamma = \{\gamma_{ij} \mid \gamma_{ij} \in \{0, 1\} \forall (i, j) \in A', \sum_{(i,j) \in A'} r_{ij} \gamma_{ij} \leq R\}$ . Following a similar set of transformations and using Lemma 2, Model 2U is seen to be equivalent to the simple minimization problem:

MODEL 1U

$$\begin{aligned}
& \min \sum_{(i,j) \in A'} u_{ij} \beta_{ij}, \\
& \text{s.t.} \quad \alpha_i - \alpha_j + \beta_{ij} + \gamma_{ij} \geq 0, \quad \forall (i, j) \in A', \\
& \quad \alpha_j - \alpha_i + \beta_{ij} + \gamma_{ij} \geq 0, \quad \forall (i, j) \in A', \\
& \quad \alpha_t - \alpha_s \geq 1, \\
& \quad \sum_{(i,j) \in A'} r_{ij} \gamma_{ij} \leq R, \\
& \quad \alpha_i \in \{0, 1\}, \quad \forall i \in N, \\
& \quad \beta_{ij}, \gamma_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A'.
\end{aligned} \tag{9}$$

### 5.5. Multiple Resources

Problems with multiple resources might occur in two cases. In the one case, we require multiple resources to interdict an arc where interdiction of arc  $(i, j)$  requires  $r_{ijl}$  units of resource  $l$  for each resource  $l \in L$ , and there are a total of  $R_l$  units of resource  $l$  available. In this case, in Model 1D, equation (2) is simply replaced by the set of equations

$$\sum_{(i,j) \in A} r_{ijl} \gamma_{ij} \leq R_l, \quad \forall l \in L. \tag{10}$$

It should be noted that if some of the  $r_{ijl}$  values were negative (suppose by interdicting some arcs, we actually capture some of the enemy's supplies) the substitution used in going from Model 4D to Model 1D would not be valid. In that case, it would be necessary to use a model of the form of Model 4D and replace equation (7) with equations (10).

In another case, it might be that the interdictor has multiple independent resources which can be used to interdict an arc. For instance, three different types of aircraft might be available to attack an arc and exactly one of the types of aircraft will be assigned to interdict the arc or none will. For this case, Model 1D can be replaced by

$$\min \sum_{(i,j) \in A} u_{ij} \beta_{ij},$$

$$\begin{aligned}
\text{s.t. } & \alpha_i - \alpha_j + \beta_{ij} + \sum_l \gamma_{ijl} \geq 0, & \forall (i, j) \in A, \\
& \alpha_t - \alpha_s \geq 1, \\
& \sum_{(i,j) \in A} r_{ijl} \gamma_{ijl} \leq R_l, & \forall l \in L, \\
& \alpha_i \in \{0, 1\}, & \forall i \in N, \\
& \beta_{ij} \in \{0, 1\}, & \forall (i, j) \in A, \\
& \gamma_{ijl} \in \{0, 1\}, & \forall (i, j) \in A, l \in L,
\end{aligned}$$

where  $\gamma_{ijl}$  is 1 if arc  $(i, j)$  is interdicted with resource  $l$  and is 0 otherwise, and, as above,  $R_l$  is the amount of resource  $l$  available.

### 5.6. Multiple Commodities

Here we consider a situation in which multiple commodities measured in the same units, such as kilograms, are being sent through the network and are of value to the enemy only in specified proportions. For instance, suppose it requires 100 kilograms of chemicals to process 1000 kilograms of coca leaves. Then, if 100 kilograms of chemicals are sent through the network along with 2000 kilograms of coca leaves, the enemy obtains the value of only 1000 kilograms of leaves.

Let  $b_k$  denote the number of units of commodity  $k$  which the enemy requires at the sink node to obtain one unit of benefit, and let  $x_{ijk}$  be the amount of flow of commodity  $k$  across arc  $(i, j)$ . The min-max problem faced by the interdictor is then

MODEL 2MC

$$\begin{aligned}
& \min_{\gamma \in \Gamma} \max_{x, v} v, \\
& \text{s.t. } v - b_k^{-1} x_{t_k s_k k} \leq 0, & \forall k, \\
& \sum_j x_{s_k j k} - \sum_j x_{j s_k k} - x_{t_k s_k k} = 0, & \forall k, \\
& \sum_j x_{ij k} - \sum_j x_{ji k} = 0, & \forall i \in N - \{s, t\}, \\
& \sum_j x_{t_k j k} - \sum_j x_{j t_k k} + x_{t_k s_k k} = 0, & \forall k, \\
& \sum_k x_{ij k} - u_{ij} (1 - \gamma_{ij}) \leq 0, & \forall (i, j) \in A, \\
& x_{ijk} \geq 0, & \forall (i, j) \in A \cup \{\cup_k \{(t_k, s_k)\}\},
\end{aligned}$$

where  $\Gamma = \{\gamma_{ij} \mid \gamma_{ij} \in \{0, 1\} \forall (i, j) \in A, \sum_{(i,j) \in A} r_{ij} \gamma_{ij} \leq R\}$ . The above problem definition allows for  $s$  and  $t$  to vary by commodity. For instance, there may only be one sink node  $t$  where the commodities are combined but the sources of the commodities may all be different. If all the sources were the same and all the sinks were the same the problem simplifies: The interdictor solves the single commodity problem, the enemy finds the maximum single commodity flow in the interdicted network, and then splits that flow among the commodities in amounts proportional to  $b_{k'}/\sum_k b_k$  for each commodity  $k'$ .

Taking the dual of the inner maximization problem of Model 2MC yields

MODEL 3MC

$$\begin{aligned}
& \min_{\gamma \in \Gamma} \min_{\alpha, \theta} \sum_{(i,j) \in A} u_{ij} (1 - \gamma_{ij}) \theta_{ij}, \\
& \text{s.t. } \alpha_{ik} - \alpha_{jk} + \theta_{ij} \geq 0, & \forall k, (i, j) \in A, \\
& \alpha_{t_k} - \alpha_{s_k} - b_k^{-1} \nu_k \geq 0, & \forall k,
\end{aligned}$$

$$\begin{aligned}
\sum_k \nu_k &\leq 1, \\
\alpha_{ik} &\geq 0, & \forall k, i \in N, \\
\theta_{ij} &\geq 0, & \forall (i, j) \in A.
\end{aligned}$$

Given an upper bound  $\bar{\theta}_{ij}$  on  $\theta_{ij}$  allows the above model to be linearized yielding the mixed integer problem:

MODEL 4MC

$$\begin{aligned}
\min \quad & \sum_{(i,j) \in A} u_{ij} \beta_{ij}, \\
\text{s.t.} \quad & \alpha_{ik} - \alpha_{jk} + \theta_{ij} \geq 0, & \forall k, (i, j) \in A, \\
& \alpha_{t_k k} - \alpha_{s_k k} - b_k^{-1} \nu_k \geq 0, & \forall k, \\
& \beta_{ij} + \bar{\theta}_{ij} \gamma_{ij} - \theta_{ij} \geq 0, & \forall (i, j) \in A, \\
& \sum_k \nu_k \leq 1, \\
& \sum_{(i,j) \in A} r_{ij} \gamma_{ij} \leq R, \\
& \theta_{ij}, \beta_{ij} \geq 0, & \forall (i, j) \in A, \\
& \gamma_{ij} \in \{0, 1\}, & \forall (i, j) \in A,
\end{aligned}$$

(Note that the interpretations of the  $\alpha$ ,  $\beta$  and  $\gamma$  variables given for Model 1D are not valid here since these variables are continuous.) In order to determine a suitable value for  $\bar{\theta}_{ij}$  for a given  $\gamma_{ij}$ , note that  $\theta_{ij}$  is the dual variable on the constraint

$$\sum_k x_{ijk} \leq u_{ij}(1 - \gamma_{ij}),$$

in Model 3MC. If the right-hand side of this equation were increased by one unit, the maximum change in the objective function would occur if this allowed one additional unit of commodity  $k'$  to flow through the network where  $k' = \operatorname{argmax}\{b_k^{-1}\}$  and this allowed  $v$  to increase by  $b_{k'}^{-1}$  units of benefit. Thus, Model 4MC is valid if  $\bar{\theta}_{ij} \equiv \max_k \{b_k^{-1}\}$ . We would use this model to solve the multicommodity variant of Problem 1 since there is no analog of Model 1D for this variant.

## 6. STRONGER FORMULATIONS

In this section, we describe valid inequalities to help tighten the LP relaxation of the basic Models 1D and 1DC and a stronger reformulation of Model 1D based on cutsets. (We note that the reformulation can be interpreted as introducing additional types of valid inequalities to Model 1D but it suffices to view the process as a reformulation.) Initial computational experience on Model 1D [1] indicates that fairly large problems can be solved without the need to introduce cuts or valid inequalities. However, experience has shown on a variety of combinatorial optimization problems, e.g., [13], that as the problems become larger and harder to solve, valid inequalities become quite useful. Thus, in anticipation of solving larger problems, we discuss two types of valid inequalities. The first type of inequality, useful for Model 1DC, is derived as a Chvatal inequality [15] with a strengthening step added. The second type of inequality, for Model 1D, is a standard knapsack inequality, e.g., [14]. The usefulness of these inequalities and the reformulation is demonstrated in the next section which gives a numerical example.

### 6.1. Type I Valid Inequalities (Model 1DC)

Consider first Model 1DC which has a cardinality constraint on the number of arcs which can be broken to minimize the maximum flow. Consider a node  $i'$  and a maximum set of arc

independent paths  $P_{i'}$  from  $i'$  to  $t$ . Assume that  $|P_{i'}| > R$  and let  $A_{i'}$  be the set of arcs on the paths  $P_{i'}$ . Then the following “Type I inequality” is valid:

$$(|P_{i'}| - R)\alpha_{i'} + \sum_{(i,j) \in A_{i'}} \beta_{ij} \geq |P_{i'}| - R. \quad (11)$$

If  $i'$  is on the  $t$  side of the optimal cut so that  $\alpha_{i'} = 1$  the constraint is redundant. However, if  $i'$  is on the  $s$  side of the cut so that  $\alpha_{i'} = 0$  the constraint simply implies that at least  $|P_{i'}|$  arcs among the paths will cross the cut and since at most  $R$  of those crossing arcs can be broken, at least  $|P_{i'}| - R$  will not be broken, i.e., at least  $|P_{i'}| - R$  of the  $\beta_{ij}$  for  $(i,j)$  on those paths must have value 1. This inequality will cut away a fractional solution (i.e., it is violated by the solution to an LP relaxation of Model 1DC) if node  $i'$  can be chosen so that  $\alpha_{i'} < 1$ ,  $|P_{i'}| > R$  and  $\beta_{ij} = 0$  for all  $(i,j) \in A_{i'}$ . Such a situation, if it exists, can be found in polynomial time by finding a maximum flow from  $i$  to  $t$  in  $G$  where the capacities of arcs are 0 if  $\beta_{ij} > 0$  and are 1 if  $\beta_{ij} = 0$ . However, there may also exist violated inequalities in which not all  $\beta_{ij}$  are 0 and these would be harder to find; an integer program could be devised to find them but this might be as hard as the original problem. It is theoretically easy to generate all type I inequalities *a priori* using a maximum flow algorithm coupled with an enumeration mechanism which would force an arc to be in a maximum flow or not. However, the number of these inequalities may be exponential in the size of  $G$  since there can be an exponential number of maximum cardinality arc independent paths from any node  $i'$  to  $t$ .

The type I valid inequality can be derived as in [15], i.e., by adding multiples of constraints together and then rounding coefficients to integers, although we add a strengthening step at the end. The constraints of Model 1DC can be written as

$$\alpha_i - \alpha_j + \beta_{ij} + \gamma_{ij} \geq 0, \quad \forall (i,j) \in A, \quad (12)$$

$$\alpha_s = 0,$$

$$\alpha_t = 1,$$

$$- \sum_{(i,j) \in A} \gamma_{ij} \geq -R, \quad (13)$$

$$\alpha_i \in \{0,1\} \quad \forall i \in N,$$

$$\beta_{ij}, \gamma_{ij} \in \{0,1\} \quad \forall (i,j) \in A.$$

As above, select a node  $i'$  and a set of paths  $P_{i'}$  with arc set  $A_{i'}$  and such that  $|P_{i'}| > R$ . With multipliers of  $1 - \epsilon$  for  $\epsilon$  sufficiently small and positive add together all constraints of the form (12) for  $(i,j) \in A_{i'}$  together with constraint (13). This yields the redundant constraint:

$$(1 - \epsilon)|P_{i'}|\alpha_{i'} - (1 - \epsilon)|P_{i'}|\alpha_t + \sum_{(i,j) \in A_{i'}} (1 - \epsilon)\beta_{ij} - \sum_{(i,j) \notin A_{i'}} (1 - \epsilon)\gamma_{ij} \geq -(1 - \epsilon)R.$$

Noting that  $\alpha_t = 1$  then yields

$$(1 - \epsilon)|P_{i'}|\alpha_{i'} + \sum_{(i,j) \in A_{i'}} (1 - \epsilon)\beta_{ij} - \sum_{(i,j) \notin A_{i'}} (1 - \epsilon)\gamma_{ij} \geq (1 - \epsilon)(|P_{i'}| - R).$$

In a standard fashion, then, all coefficients on the left-hand side can be rounded up to integer values while maintaining validity of the inequality. Then, since the left-hand side is integer in an optimal solution the right-hand side can be rounded up to the nearest integer yielding:

$$|P_{i'}|\alpha_{i'} + \sum_{(i,j) \in A_{i'}} \beta_{ij} \geq |P_{i'}| - R. \quad (14)$$

If  $\alpha_{i'} = 0$  in an optimal solution the coefficient on  $\alpha_{i'}$  is irrelevant. On the other hand, if  $\alpha_{i'} = 1$  constraint (14) is redundant and remains so if the coefficient on  $\alpha_{i'}$  is decreased to  $|P_{i'}| - R$  which yields the type I inequality (11). This inequality is clearly stronger than (14).

The type I valid inequality can be extended to Model 1D where equations (13) are replaced by

$$- \sum_{(i,j) \in A} r_{ij} \gamma_{ij} \geq -R. \quad (15)$$

However, in forming the inequality, a good choice for the multiplier on constraint (15) is not obvious. Rather than pursuing this subject, we consider completely different inequalities for Model 1D.

### 6.2. Type II Valid Inequalities (Model 1D)

Considering constraint (2) alone in Model 1D, it is clear that any valid inequality for a knapsack problem defined on that constraint is valid for the full model. Following the development in [12] suppose that  $A' \subseteq A$  is such that  $\sum_{(i,j) \in A'} r_{ij} > R$  but  $\sum_{(i,j) \in A''} r_{ij} \leq R$  for any  $A'' \subset A'$ . Then,  $A'$  is a *minimal independent set*. The *extension* of  $A'$ , denoted  $E(A')$  is defined by  $E(A') = A' \cup \{(i', j') \in A - A' \mid r_{i'j'} \geq r_{ij} \forall (i, j) \in A'\}$ . Then, for any minimal independent set  $A'$ , the following inequality is valid for Model 1D:

$$\sum_{(i,j) \in E(A')} r_{ij} \gamma_{ij} \leq |A'| - 1. \quad (16)$$

We refer to this inequality as a type II valid inequality. The reader is referred to [12] for a discussion of other types of knapsack inequalities which could be applied to Model 1D.

### 6.3. A Cutset-Based Reformulation (Model 1D)

Here we discuss a cutset-based reformulation of Model 1D. If all cutsets in graph  $G$  were enumerated and there were not too many, solution of Model 1D would not be trivial but it would be relatively easy: Let  $A^C$  be the set of forward arcs in cutset  $C$ . Solve the knapsack problem on each cutset

$$\begin{aligned} \max \quad & \sum_{(i,j) \in A^C} u_{ij} \gamma_{ij}, \\ \text{s.t.} \quad & \sum_{(i,j) \in A^C} r_{ij} \gamma_{ij} \leq R, \\ & \gamma_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A^C, \end{aligned}$$

to obtain solution  $\gamma^C$  and define remaining capacity to be  $U^C = \sum_{(i,j) \in A^C} (1 - \gamma_{ij}^C) u_{ij}$ . Then, the optimal cutset is that cutset which has the minimum value of  $U^C$  and the optimal arcs to break are those arcs  $(i, j)$  such that  $\gamma_{ij}^C = 1$ .

To obtain cutset-based valid inequalities for Model 1D, however, we need to develop a cutset-based, mathematical programming formulation of the model. Letting  $\mathcal{C}$  denote the set of all  $s$ - $t$  cutsets in  $G$ , the following formulation is probably the most obvious:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} u_{ij} \beta_{ij}, \\ \text{s.t.} \quad & x_C - \gamma_{ij} - \beta_{ij} \leq 0, \quad \forall (i, j) \in A^C, \end{aligned} \quad (17)$$

$$\sum_{C \in \mathcal{C}} x_C = 1, \quad (18)$$

$$\begin{aligned} \sum_{(i,j) \in A} r_{ij} \gamma_{ij} &\leq R, \\ \gamma_{ij}, \beta_{ij} &\in \{0, 1\}, \quad \forall (i, j) \in A, \\ x_C &\in \{0, 1\}, \quad \forall C \in \mathcal{C}, \end{aligned}$$



where  $\gamma_{ij}$  and  $\beta_{ij}$  are defined as before and  $x_C = 1$  if  $C$  is the optimal cut and otherwise,  $x_C = 0$ . The difficulty with this formulation is that it is not possible to write down a simple, useful relaxation of it. If constraint (18) is deleted the optimal solution is 0. If constraint (18) is included but constraints (17) are deleted for some  $C$ , the optimal solution is also 0. The following formulation, called Model 1C, avoids this problem.

Model 1C uses  $\gamma_{i,j}$ ,  $\beta_{ij}$  and  $\alpha_i$  defined as in Model 1D. With respect to a cutset  $C$ , let  $N_s^C$  denote the nodes on the  $s$  side of the cutset and let  $N_t^C$  denote the nodes on the  $t$  side of the cut. New binary variables  $\delta_s^C$  and  $\delta_t^C$  are defined so that  $\delta_s^C = 0$  and  $\delta_t^C = 1$  if and only if  $C$  is the optimal cutset. Finally, let  $R^C$  denote the amount of resource consumed in the solution to the knapsack problem defined with respect to cutset  $C$ , let  $A_\gamma^C$  be the set of arcs  $(i, j)$  such that  $\gamma_{ij} = 1$  in that knapsack solution and let  $A_\beta^C$  be the set of arcs  $(i, j)$  such that  $\beta_{ij} = 1$  in that knapsack solution. The reformulation is

MODEL 1C

$$\begin{aligned}
\min \quad & \sum_{(i,j) \in A} u_{ij} \beta_{ij}, \\
\text{s.t.} \quad & \alpha_i - \alpha_j + \beta_{ij} + \gamma_{ij} \geq 0, & \forall (i, j) \in A, \\
& \delta_s^C - \sum_{i \in N_s^C} \alpha_i \leq 0, & \forall C \in \mathcal{C}, \\
& \delta_t^C - \sum_{j \in N_t^C} \alpha_j \geq -|N_t^C| + 1, & \forall C \in \mathcal{C}, \\
& \delta_t^C - \delta_s^C - \gamma_{ij} \leq 0, & \forall (i, j) \in A_\gamma^C, C \in \mathcal{C}, \\
& \delta_t^C - \delta_s^C - \beta_{ij} \leq 0, & \forall (i, j) \in A_\beta^C, C \in \mathcal{C}, \\
& \sum_{(i,j) \in A} r_{ij} \gamma_{ij} + \sum_{C \in \mathcal{C}} (R - R^C) \delta_t^C - \sum_{C \in \mathcal{C}} (R - R^C) \delta_s^C \leq R, \\
& \alpha_s = 0, \\
& \alpha_t = 1, \\
& \text{All variables} \in \{0, 1\}.
\end{aligned}$$

Note that the model has been formulated so that it is valid even if  $\mathcal{C}$  is replaced by some subset of  $\mathcal{C}$ . In fact, if  $\mathcal{C} = \emptyset$  the model reverts to Model 1D.

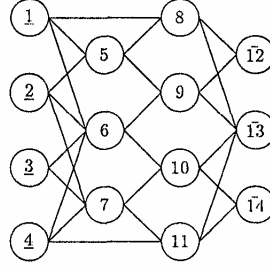
To see that the formulation is stronger than Model 1D (we do not prove it in general) consider a solution  $(\gamma^*, \beta^*, \alpha^*)$  to the LP relaxation of Model 1D in which all variables are optimal to the integer program (IP) except one  $\beta_{i^*, j^*}^*$ , and the paired  $\gamma_{i^*, j^*}^*$  are fractional, i.e., the LP solution would be optimal to the IP if  $\gamma_{i^*, j^*}^*$  were forced to 0 and  $\beta_{i^*, j^*}^*$  were forced to 1. Such solutions, in which the optimal cut  $C^*$  is identified but not proven optimal, can and do occur in practice. In one of these LP solutions the resource constraint will be tight although it will be slack in the solution to the IP. Because of this, if Model 1C is formulated with  $\mathcal{C} = \{C^*\}$ , it is easy to see that the LP solution will not be feasible (the modified resource constraint will be violated) for the new formulation and thus Model 1C is stronger than Model 1D.

## 7. A NUMERICAL EXAMPLE

In this section, we compare our methodology to that of [4] using the example given in that paper. The test network, which we denote  $G = (N, A)$ , is given in Figure 2 and the capacity and resource necessary to break each arc is given in Table 1. A total of 15 units of resource is available for interdicting arcs.

$G$  is undirected and it has multiple sources  $N^S = \{1, 2, 3, 4\}$  and multiple sinks  $N^T = \{12, 13, 14\}$ . We use a hybrid of Models 1D and 1U to solve this problem with  $\alpha_i$  replaced by 0 for all  $i \in N^S$  and  $\alpha_i$  replaced by 1 for all  $i \in N^T$ .

Let  $A$  be partitioned into  $A_{ST}$  and  $\bar{A}_{ST}$  where  $A_{ST}$  denotes those arcs incident to a node in  $N^S$  or  $N^T$  and  $\bar{A}_{ST}$  denotes the complementary set. Arcs in  $A_{ST}$  may be replaced by directed



( $\underline{j}$  indicates  $j$  is a source node,  $\bar{j}$  indicates  $j$  is a demand node.)

Figure 2. Sample network.

Table 1. Network data.

Arc	Capacity	Resource	Arc	Capacity	Resource
(1,5)	60	5	(6,9)	120	4
(1,8)	70	4	(6,10)	150	6
(1,6)	60	5	(7,10)	120	6
(2,5)	50	3	(7,11)	80	4
(2,6)	50	3	(8,12)	80	4
(2,7)	60	5	(8,13)	50	5
(3,6)	100	3	(9,12)	100	5
(3,7)	80	5	(9,13)	80	4
(4,6)	50	5	(10,13)	180	6
(4,7)	100	5	(10,14)	100	4
(4,11)	80	4	(11,13)	80	5
(5,8)	60	4	(11,14)	100	6
(5,9)	60	7			

arcs: There need not be any flow into a source node so any arc  $(i, j)$  with  $i \in N^S$  may be replaced by a directed arc  $(i, j)$ ; likewise, any arc  $(i, j)$  with  $j \in N^T$  may be replaced by a directed arc  $(i, j)$ . Let  $\bar{A}'_{ST}$  be defined from  $\bar{A}_{ST}$  as  $A'$  is defined from  $A$ .

The model we solve then is

$$\begin{aligned}
 \min \quad & \sum_{(i,j) \in A_{ST}^d \cup \bar{A}'_{ST}} u_{ij} \beta_{ij}, \\
 \text{s.t.} \quad & \alpha_i - \alpha_j + \beta_{ij} + \gamma_{ij} \geq 0, & \forall (i,j) \in A_{ST}^d, \\
 & \alpha_i - \alpha_j + \beta_{ij} + \gamma_{ij} \geq 0, & \forall (i,j) \in \bar{A}'_{ST}, \\
 & \alpha_j - \alpha_i + \beta_{ij} + \gamma_{ij} \geq 0, & \forall (i,j) \in \bar{A}'_{ST}, \\
 & \alpha_i \equiv 0, & \forall i \in N^S, \\
 & \alpha_i \equiv 1, & \forall i \in N^T, \\
 & \sum_{(i,j) \in A_{ST}^d \cup \bar{A}'_{ST}} r_{ij} \gamma_{ij} \leq R, \\
 & \alpha_i \in \{0, 1\}, & \forall i \in N, \\
 & \beta_{ij}, \gamma_{ij} \in \{0, 1\} \forall, & \forall (i,j) \in A'.
 \end{aligned}$$

The data in Table 1 yields a model with 31 constraints and 43 variables. The network allows 720 units of flow if no arcs are interdicted. The optimal solution to the problem was obtained using the branch-and-bound algorithm within LINDO [16] and required eight branches versus the

sixteen required by Ghare *et al.* The initial LP relaxation has a value of 320 while the optimal solution, obtained on the third branch, has 340 units of flow remaining. The interdicted arcs are (6, 9), (10, 13), and (10, 14), while the unbroken arcs which define the minimum capacity cut are (1, 8), (1, 5), (2, 5), (4, 11) and (7, 11).

As asserted in the previous section, the solution to the LP relaxation of the above problem might be all integer, except for one  $\beta_{ij}$  and its cohort  $\gamma_{ij}$  being fractional. The  $\alpha_i$  in this instance would all be 0 or 1 and identify a cut which might be the optimal cut. Such is the case for this problem. Adding inequalities of type II and reformulating the problem based on the identified cut results in an improvement in the value of the LP relaxation to 328.3. The branch-and-bound algorithm then requires 3 branches to solve the problem optimally.

To test inequalities of type I, the resource constraint in the above problem was replaced by a cardinality constraint. However, the solution to the LP relaxation of this model is integral and solves the IP irrespective of what integer value for  $R$  is used. Consequently, it was necessary to modify the  $u_{ij}$  until a fractional LP solution was found; a value of  $R = 3$  was used. Evaluating the LP solution yielded a type I inequality based on four arc-independent paths running from node 9 to the sink nodes. The LP relaxation of the model proved to be integer optimal after adding this one inequality.

## 8. CONCLUSIONS

This paper has described a simple network interdiction model and its variants in which an interdicator, using limited resources, interdicts arcs in a capacitated network so as to minimize the maximum flow that can be pushed through the network by an adversary. The basic problem is shown to be NP-complete even when the interdiction of an arc requires exactly one unit of resource. A new integer programming model is developed for the basic problem and is shown to be easily modified and extended to handle variants and generalizations of this problem. This is in contrast to the methods previously known which cannot handle any generalizations such as multiple resource constraints or are restricted to planar networks.

In anticipation of solving large network interdiction problems, valid inequalities were developed to tighten the LP relaxation of the integer programming models. A cutset-based reformulation of the problem was also developed. A small computational example illustrates the use and flexibility of the basic model and shows that the valid inequalities and the reformulation can achieve quicker solutions.

## REFERENCES

1. R. Steinrauf, A network interdiction model, M.S. Thesis, Naval Postgraduate School, (May 1991).
2. L.R. Ford and D.R. Fulkerson, *Flows in Networks*, Princeton University, Princeton, NJ, (1962).
3. A.W. McMasters and T.M. Mustin, Optimal interdiction of a supply network, *Naval Research Logistics Quarterly* **17** (3), 261–268 (September 1970).
4. P.M. Ghare, D.C. Montgomery and W.C. Turner, Optimal interdiction policy for a flow network, *Naval Research Logistics Quarterly* **18**, 37–45 (1971).
5. R.L. Helmbold, A counter capacity network interdiction model, R-611-PR, Rand Corporation, Santa Monica, CA, March 1971, **17** (4), 37–45 (1971).
6. R.D. Wollmer, Removing arcs from a network, *Journal of the Operations Research Society of America* **12**, 934–940 (1964).
7. E.P. Durbin, An interdiction model of highway transportation, RM-4945-PR, Rand Corporation, Santa Monica, CA, (May 1966).
8. S.H. Lubore, H.D. Ratliff and G.T. Sicilia, Determining the most vital link in a flow network, *Naval Research Logistics Quarterly* **18** (4) (December 1971).
9. C.P. Preston, Interdiction of a transportation network, M.S. Thesis, Naval Postgraduate School, Monterey, CA, (March 1972).
10. H.D. Ratliff, G.T. Sicilia and S.H. Lubore, Finding the  $n$  most vital links in flow networks, *Management Science* **21** (5), 531–539 (January 1975).
11. M.R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman and Co., San Francisco, (1979).
12. G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley and Sons, Inc., New York, (1988).
13. H.P. Crowder, M.W. Padberg and E.L. Johnson, Solving large-scale zero-one linear programming problems, *Operations Research* **31**, 803–834 (1983).

14. E. Balas, Facets of the knapsack polytope, *Mathematical Programming* **8**, 146–164 (1975).
15. V. Chvatal, Edmonds polytopes and a hierarchy of combinatorial problems, *Discrete Mathematics* **4**, 305–337 (1973).
16. L. Schrage, *Linear, Integer and Quadratic Programming with LINDO*, The Scientific Press, Palo Alto, CA, (1986).